**\*\*NOTE: Some steps are under the assumption that a brief overview of xAPI or the xAPI Spec have been presented. It’s not necessary to do this beforehand, but in some of the steps more details might have to be given about xAPI. You can also login as lou/password if you want data to show up in the My Account page while demoing**

**\*Open** [**https://lrs.adlnet.gov**](https://lrs.adlnet.gov) **on screen**

The ADL LRS is an example implementation of the server side component of the xAPI. It is an open source project hosted on ADL’s GitHub site.

**\*Open** [**https://github.com/adlnet/ADL\_LRS**](https://github.com/adlnet/ADL_LRS)

Organizations are welcome to download, update and contribute to the project. We have a detailed readme file to walk you through the setup and a few wiki pages that go into further implementation and installation details. **\*Open** [**https://github.com/adlnet/ADL\_LRS/wiki**](https://github.com/adlnet/ADL_LRS/wiki) **and show a few of the pages**

**\*Back to** [**https://lrs.adlnet.gov**](https://lrs.adlnet.gov)

It fully implements all of the requirements as defined in the xAPI spec for the server side component. This is the hosted version that is available for the public to try out and to use while developing content. Occasionally the developers take down the LRS for maintenance and may need to clear the statements so please don’t send anything sensitive.

Also on the LRS, we have links to the original xAPI prototypes that were created with the first release of the spec and the open source code for the examples. **\*Click Original GitHub Examples links just to show they’re there**.  
  
We also have a link to our statement viewer that will show you the current statements stored in the LRS. Normally an LRS would be locked down and secure so you could only see statements posted by yourself or your organization, but we leave it open to view all statements for learning and demonstration purposes. **\*Click Launch for the Statement Viewer and show off statements, click on one to show the JSON**. All of the statements are listed and parsed to human-readable format, almost like an activity stream. If you click one of the statements it’ll show you the actual JSON data that is being sent and stored inside the LRS.

**\*Click on Statement Validator**

Attached to the LRS is a simple statement validator. Here you can copy and paste your statement in JSON format and the LRS will respond with any errors in it so it can be fixed. **\*Paste this blob to show it’s valid, then remove actor to show it’s invalid:**{

"actor": {

"mbox": "mailto:test@example.com",

"objectType": "Agent"

},

"object": {

"id": "8f87ccde-bb56-4c2e-ab83-44982ef22df0",

"objectType": "StatementRef"

},

"verb": {

"display": {

"en-US": "commented"

},

"id": "http://example.com/commented"

}

}

**\*Click xAPI Tools**

Here are all of the open source xAPI tools ADL has created. The statement viewer we went over already is here, as well as JavaScript and Java xAPI libraries. There are a few examples that implement those libraries and also a lab that explores parts of the statement a bit more.

**\*Click Android App**  
There is also a basic xAPI-enabled app available in the Google Play Store for Android devices. It’s a simple SCORM-y app that sends user actions to the LRS via the xAPI statements API and activity state API.

**\*Scroll down the LRS homepage**

The LRS has two types of authentication, HTTP basic and OAuth. To sign up with basic authentication, you just need to create a username and password and supply an email. Every request sent to the LRS on your behalf would include your username and password base64 encoded in the authentication header. If you would like an application to act on your behalf, you would use your username and password to login, and add an application to your account. After the OAuth process completes, you’d have your token and secret key to use.  
  
\***If you need the participants to make their own accounts do it now**  
Please sign up now on the LRS. Create a unique username and password and remember that all information on here is available to everyone so we don’t recommend using important email addresses or sensitive passwords or sensitive data in the statements.

**\*After they login, show the user’s homepage**  
Here is what the authority of the statement will look like if you submit any. It’ll have your username and email address listed in the agent. Also here, you can view all of your statements, activity states, web hooks, and OAuth information. **\*Click Statement and Activity States just to show they’d be there**

**\*Click the OAuth tabs**

Just to show you where your OAuth information is, you can activate/deactivate your apps and once you receive access tokens for them, view the scope permissions allowed for it.

**\*Click Webhooks**

We added webhooks to our LRS to allow users ‘forwarding’ of their statements stored. This is outside the scope of the xAPI spec so you don’t see anything about this listed anywhere else. Essentially as statements come in, the LRS will look through all of its registered webhooks to see if it meets the criteria listed here in the filters. If the statements meet the criteria the LRS will attempt to send them to the listed endpoint for further processing.